



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,168	07/09/2001	Akira Kamiya	2001_0976A	5407
513 7590 06/26/2008 WENDEROTH, LIND & PONACK, L.L.P. 2033 K STREET N. W. SUITE 800 WASHINGTON, DC 20006-1021				
EXAMINER				
RAO, ANAND SHASHIKANT				
ART UNIT		PAPER NUMBER		
2621				
MAIL DATE		DELIVERY MODE		
06/26/2008		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte AKIRA KAMIYA

Appeal 2008-0517
Application 09/900,168¹
Technology Center 2600

Decided: June 26, 2008

Before KENNETH W. HAIRSTON, MAHSHID D. SAADAT,
and SCOTT R. BOALICK, *Administrative Patent Judges*.

BOALICK, *Administrative Patent Judge*.

¹ Application filed July 9, 2001. Application 09/900,168 claims the benefit under 35 U.S.C. § 119 of Japanese patent application 2000-208074, filed July 10, 2000. The real party in interest is Matsushita Electric Industrial Co., Ltd.

DECISION ON APPEAL

This is an appeal under 35 U.S.C. § 134(a) from the final rejection of claims 3, 4, 7, and 8, all the claims pending in the application. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

STATEMENT OF THE CASE

Appellant's invention relates to a multiple decoding apparatus and method for reproducing data designated from an MPEG (Moving Picture Experts Group) transport stream of encoded data. (Spec. 1:6-10.)

Claim 3 is exemplary:

3. A multiple decoding apparatus for receiving a broadcasting signal composed of a plurality of encoded data and for simultaneously decoding two or more of the encoded data, the multiple decoding apparatus comprising:

a reproduction controller for outputting control information related to decoding and reproduction of data;

a data extractor for receiving the broadcasting signal and extracting at least audio data and video data which are designated by the control information;

a buffer for storing at least the audio data and the video data extracted by said data extractor;

a buffer manager for controlling said buffer in accordance with the control information for said buffer;

a data flow controller for distributing at least the audio data and the video data stored in said buffer for each data type

and transferring at least the audio data and the video data in accordance with provided transfer conditions;

a plurality of separate buffers for respectively storing at least the audio data and the video data distributed and transferred by said data flow controller according to each data type;

a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in said plurality of separate buffers so as to be associated with each other in accordance with information for specifying said plurality of separate buffers;

a plurality of decoders respectively corresponding to said plurality of separate buffers for decoding at least the audio data and the video data stored in said plurality of separate buffers and outputting two or more decoded data; and

a decoding controller for selecting a separate buffer and a decoder, which are used for the decoding, according to a usage status of said decoder from among said plurality of separate buffers and said plurality of decoders in accordance with the control information, and outputting information related to said separate buffer selected by said decoding controller, the transfer conditions based on said separate buffer selected by said decoding controller, and an instruction to start decoding, respectively, to said separate buffer manager, said data flow controller, and said decoder selected by said decoding controller, wherein

said separate buffer manager outputs, when a specific separate buffer becomes full of data, an overflow notification that said specific separate buffer overflows to said decoding controller,

said decoding controller outputs, upon receipt of the overflow notification that said specific separate buffer overflows, an instruction to stop data transfer to said specific

separate buffer to said data flow controller, outputs an instruction to stop decoding to a decoder corresponding to said specific separate buffer, and outputs an instruction to initialize said specific separate buffer to said separate buffer manager,

said separate buffer manager initializes said specific separate buffer in accordance with the instruction to initialize said specific separate buffer from said decoding controller without initializing said buffer, and

the multiple decoding apparatus resumes all processing which was stopped as a result of said specific separate buffer becoming full after said specific separate buffer is initialized.

The prior art relied upon by the Examiner in rejecting the claims on appeal is:

Haskell	US 5,159,447	Oct. 27, 1992
Siong	US 6,028,632	Feb. 22, 2000
Kawakami	US 6,332,058 B1	Dec. 18, 2001 (filed Mar. 18, 1998)

Claims 3, 4, 7, and 8 stand rejected under 35 U.S.C. § 103(a) as being obvious over Kawakami, Siong, and Haskell.

Rather than repeat the arguments of Appellant or the Examiner, we make reference to the Brief and the Answer for their respective details. Only those arguments actually made by Appellant have been considered in this decision. Arguments that Appellant did not make in the Brief have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(vii).

ISSUE

The issue is whether Appellant has shown that the Examiner erred in rejecting the claims under 35 U.S.C. § 103(a). The issue turns on whether Kawakami, Siong, and Haskell teach or suggest: (1) a separate buffer manager that outputs an overflow notification to a decoding controller when a specific separate buffer becomes full of data; (2) a decoding controller that, upon receipt of the overflow notification, outputs an instruction to a data flow controller to stop data transfer to the specific separate buffer, outputs an instruction to the decoder corresponding to the specific separate buffer to stop decoding, and outputs an instruction to the separate buffer manager to initialize the specific separate buffer; (3) a separate buffer manager that initializes the specific separate buffer according to the instruction from the decoding controller without initializing "said buffer" [i.e., a buffer --which is different than the "separate buffer -- that stores audio data and video data extracted by a data extractor]; and (4) resuming processing that was stopped as a result of the specific separate buffer becoming full after the specific separate buffer is initialized.

PRINCIPLES OF LAW

All timely filed evidence and properly presented arguments are considered by the Board in resolving an obviousness issue on appeal. *See In re Piasecki*, 745 F.2d 1468, 1472 (Fed. Cir. 1984).

In the examination of a patent application, the Examiner bears the initial burden of showing a prima facie case of unpatentability. *Id.* at 1472. When that burden is met, the burden then shifts to the Applicant to rebut. *Id.*; *see also In re Harris*, 409 F.3d 1339, 1343-44 (Fed. Cir. 2005) (finding

rebuttal evidence unpersuasive). If the Applicant produces rebuttal evidence of adequate weight, the prima facie case of unpatentability is dissipated. *In re Piasecki*, 745 F.2d at 1472. Thereafter, patentability is determined in view of the entire record. *Id.* However, on appeal to the Board it is the Appellant's burden to establish that the Examiner did not sustain the necessary burden and to show that the Examiner erred.

"Section 103 forbids issuance of a patent when 'the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.'" *KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1734 (2007).

ANALYSIS

Appellant contends that the Examiner erred in rejecting claims 3, 4, 7, and 8 as being obvious over Kawakami, Siong, and Haskell. Reviewing the record before us, we agree. In particular, we find that Appellant has shown that the Examiner failed to make a prima facie showing of obviousness with respect to claims 3, 4, 7, and 8.

Regarding claim 3, we agree with Appellant (Br. 12-13) that the applied references do not teach or suggest a separate buffer manager and a decoding controller, as claimed. In particular, neither Kawakami, Siong, nor Haskell, alone or in combination, teach or suggest: (1) a separate buffer manager that outputs an overflow notification to a decoding controller when a specific separate buffer becomes full of data; (2) a decoding controller that, upon receipt of the overflow notification, (a) outputs an instruction to a data

flow controller to stop data transfer to the specific separate buffer, (b) outputs an instruction to the decoder corresponding to the specific separate buffer to stop decoding, and (c) outputs an instruction to the separate buffer manager to initialize the specific separate buffer; (3) a separate buffer manager that initializes the specific separate buffer according to the instruction from the decoding controller without initializing "said buffer" [i.e., a buffer --which is different than the "separate buffer -- that stores audio data and video data extracted by a data extractor]; and (4) resuming processing that was stopped as a result of the specific separate buffer becoming full after the specific separate buffer is initialized, as recited by claim 3.

The Examiner found that Haskell teaches items 1, 2(a), and 2(b) above because, according to the Examiner, Haskell:

discloses a buffer control for [a] variable bit rate channel . . . and teaches the conventional notification of overflow situations associated with encoder and decoder buffers . . . and the particular termination of packets of data within the decoder as one way of *preventing* overflow in the buffers, thereby stopping decoding to the decoder, data extraction, data transfer to the specific buffer, and discarding [the] data directed toward the specific buffer.

(Ans. 8 (emphasis added.)) The Examiner then "noted that Haskell et al is however silent as to the initialization of the specific separate buffer in response to the overflow notification without initializing the buffer and the subsequent resuming of the processing [i.e., items 2(c), 3, and 4 above]."

(Ans. 8.) Nevertheless, the Examiner reasoned that:

it is considered obvious even without specific disclosure that once the packets are terminated within Haskell due to buffer overflow, the specific buffers of Haskell must be initialized

since the existing data within the buffers are of no use and so that the buffers could be properly re-set. Such buffer initialization specifics as taught by Haskell may certainly be provided within Kawakami wherein the specific separate buffers 34 of Kawakami may also be initialized in response to an overflow notification. And it is considered obvious that since it is only necessary for the specific separate buffers 34 within Kawakami to be initialized in response to an overflow situation, the initialization of buffers 30 within Kawakami is obviously not necessary. Further, after such buffer initialization and re-setting within Haskell, all processing will therefore be resumed, and the discarded data is released (i.e., the existing data in the buffer is of no use and therefore is released) after buffer initialization.

(Ans. 8.) We do not agree.

Instead, we agree with Appellant (Br. 13) that Haskell is directed to overflow prevention rather than recovery after an overflow has occurred. The claim recites that, "when a specific separate buffer become full of data, an overflow notification that said specific separate buffer overflows [is output by the separate buffer manager] to said decoding controller." In other words, the claim requires that the overflow notification be generated *after* an overflow has occurred.

The Examiner cites the teaching at column 16, lines 28-30 of Haskell as teaching recovery from overflow -- i.e., after an overflow has occurred. The cited portion of Haskell teaches that "[o]ther actions are also possible to alleviate overflow of [the] decoder buffer." However, when read in context, the cited portion of Haskell teaches alternate methods of overflow prevention rather than methods of recovery from an overflow that has already occurred. In particular, the preceding sentence reads "[i]n this embodiment, *if overflow threatens*, encoder 101-1 is simply told to reduce its

output data rate." (Haskell, col. 16, ll. 26-27 (emphasis added).) Thus, in context, it is clear that the "other actions" described at col. 16, lines 30-40, refer to actions to prevent overflow. The Examiner has not pointed to, nor do we find, any other portion of Haskell that teaches or suggests recovery from an overflow. Instead, we find that Haskell is directed exclusively to preventing an overflow from occurring. (*See, e.g.*, Haskell Abstract; col. 1, ll. 6-8, 28-29, 32-33, 41-43, 64-67; col. 2, ll. 5-12, 41-43; col. 4, ll. 63-67; col. 5, ll. 12-16, 47-51; col. 6, ll. 8-9, 14; col. 7, ll. 33-35, 67; col. 8, ll. 42-43, 53-54; col. 11, ll. 64-66; col. 13, ll. 17-18; col. 14, ll. 12-14, 63-64; col. 16, ll. 7-9, 25-39; col. 17, ll. 67-68; col. 18, ll. 7-13, 17-19.) Therefore, Haskell does not teach items 1, 2(a), and 2(b) above.

In addition, as the Examiner admits (Ans. 8), Haskell is silent as to items 2(c), 3, and 4 listed above regarding the initialization of the specific separate buffer and resumption of processing recited by claim 3. We find nothing in Haskell to suggest these limitations. Nor do we agree that these limitations are obvious without specific disclosure or other evidence being presented. For example, we find nothing to teach or suggest the recited interaction between the "separate buffer" and "said buffer" -- i.e., initializing the separate buffer "without initializing said buffer." In addition, there is no evidence before us to show that these limitations are a predictable variation of the prior art. Nor is there evidence before us to show that these limitations would be common sense or a creative step that a person of ordinary skill in the art would employ.

Therefore, we find that Haskell does not teach or suggest the separate buffer manager or the decoding controller as recited by claim 3. Kawakami and Siong do not remedy these deficiencies of Haskell. Claims 4, 7, and 8

each recites limitations similar to those we found lacking in the applied references with respect to claim 3.

Accordingly, we conclude that Appellant has shown that the Examiner erred in rejecting claims 3, 4, 7, and 8 under 35 U.S.C. § 103(a).

CONCLUSION OF LAW

We conclude that Appellant has shown that the Examiner erred in rejecting claims 3, 4, 7, and 8 for obviousness under 35 U.S.C. § 103.

DECISION

The rejection of claims 3, 4, 7, and 8 for obviousness under 35 U.S.C. § 103 is reversed.

REVERSED

KIS

WENDEROTH, LIND & PONACK, L.L.P.
2033 K STREET N.W.
SUITE 800
WASHINGTON, DC 20006-1021